Advanced Health Monitoring of Computer Clusters

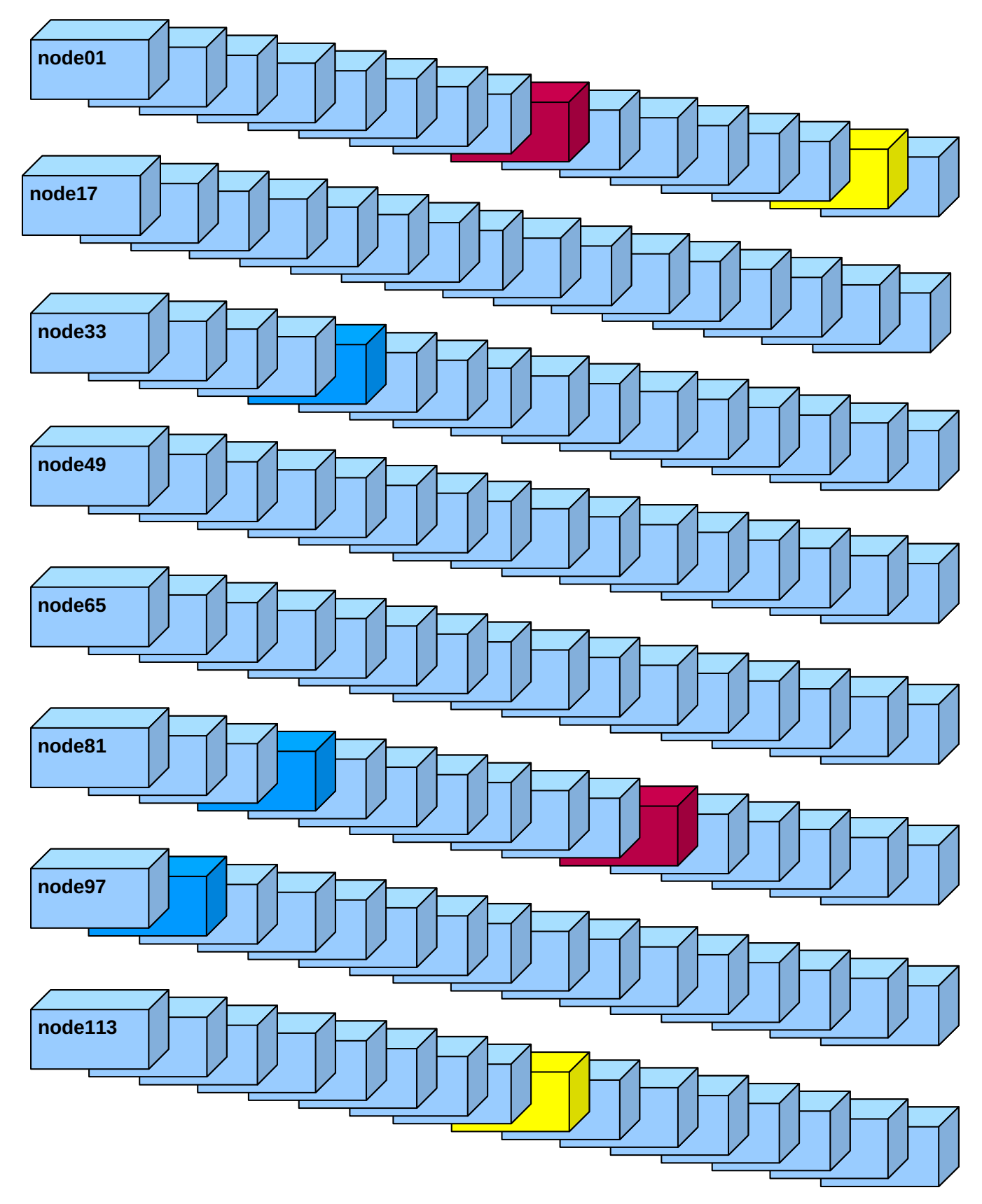
Dylan Merrigan, Caleb Morse, Sherry Salas
Mentors: Susan Coulter, Andree Jacobson, Kevin

Abstract:
As the number of nodes and processors in a cluster grows, the amount of possible system errors drastically increases. The addition of nodes/processors to a cluster makes the system's interoperability and networking more complex. To ensure accuracy and speed of a cluster, it is important the system be monitored and managed in an efficient and effective manner. Monitoring tools must recognize failures and performance degradation in the cluster and report them to system administrators in a timely manner to avoid smaller issues turning into system-wide problems. This must be done quickly and accurately without taking resources away from the system. Our solution to this problem is to use Nagios and Cacti, along with tools we have designed and written to monitor the system. Failures will be injected into the system to ensure monitoring tools are providing accurate data.

Problem:
As a cluster grows in size it becomes much more difficult to monitor

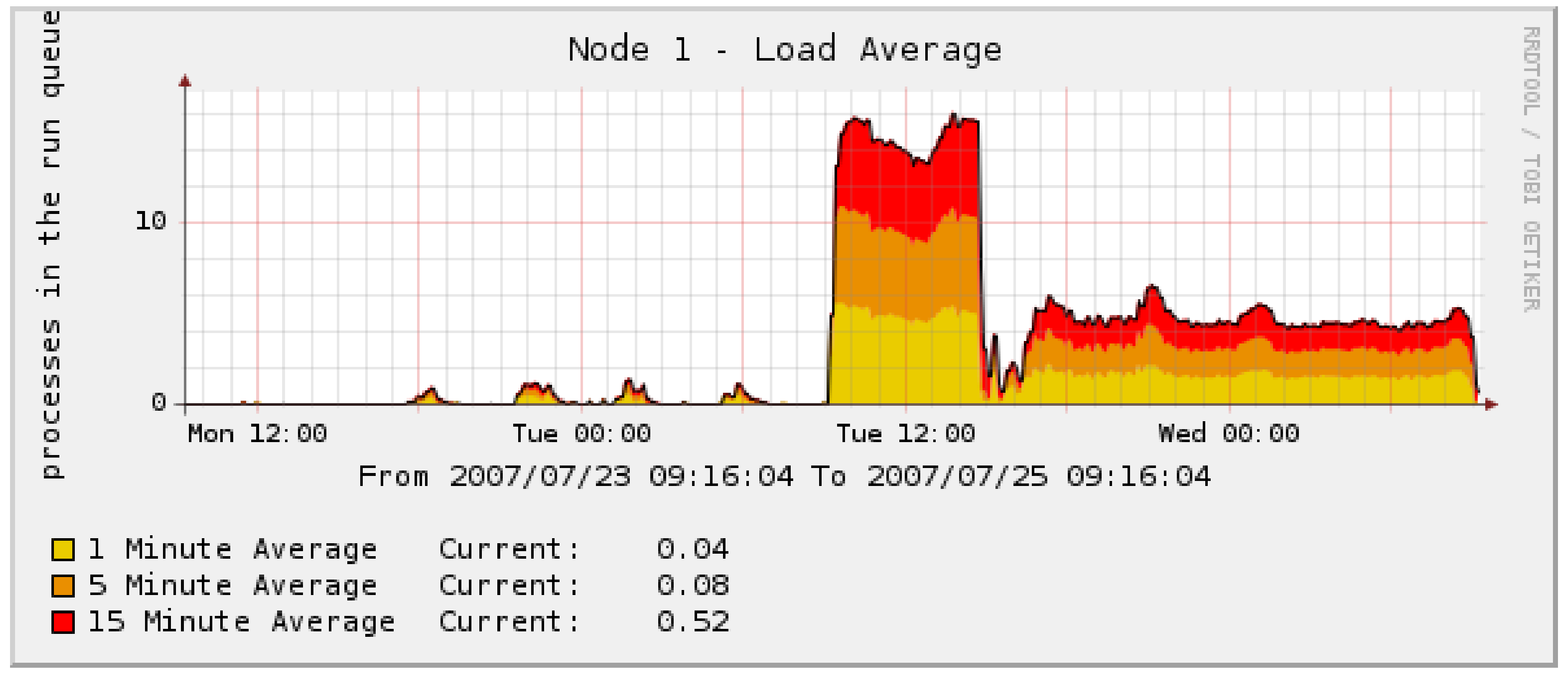
Mean time to failure increases dramatically with a large cluster

This is an image of a 128 node cluster. This representation helps demonstrate how difficult it is to monitor large clusters. 128 nodes is relatively small for one segment of a d and most clusters have multiple segments. Additionally, each of these nodes may contain several processors (the roadrunner phase 1 systems have 8) and multiple interface cards. The networking connecting these nodes is also a critical element which must be monitored.



The Monitors:

Nagios Monitoring:
Scalability- has been an effective tool in our small 10 node cluster, however the use of Nagios in a large scale cluster would not be effective
Robustness-
Extensibility- allows extension of services. We were able to write new script and implement them in our system.
Manageability-
Portability- used with Ubuntu we encountered problems
Overhead- must ssh to each node when a service is checked. This creates high overhead



🟨 1 Minute Average Current: 0.04
🟧 5 Minute Average Current: 0.08
🟥 15 Minute Average Current: 0.52

Solution:

Monitor and manage the system
- Identify system faults
- Recognize performance degradation
- Prompt notifications
- Low overhead on compute nodes
- Affordable

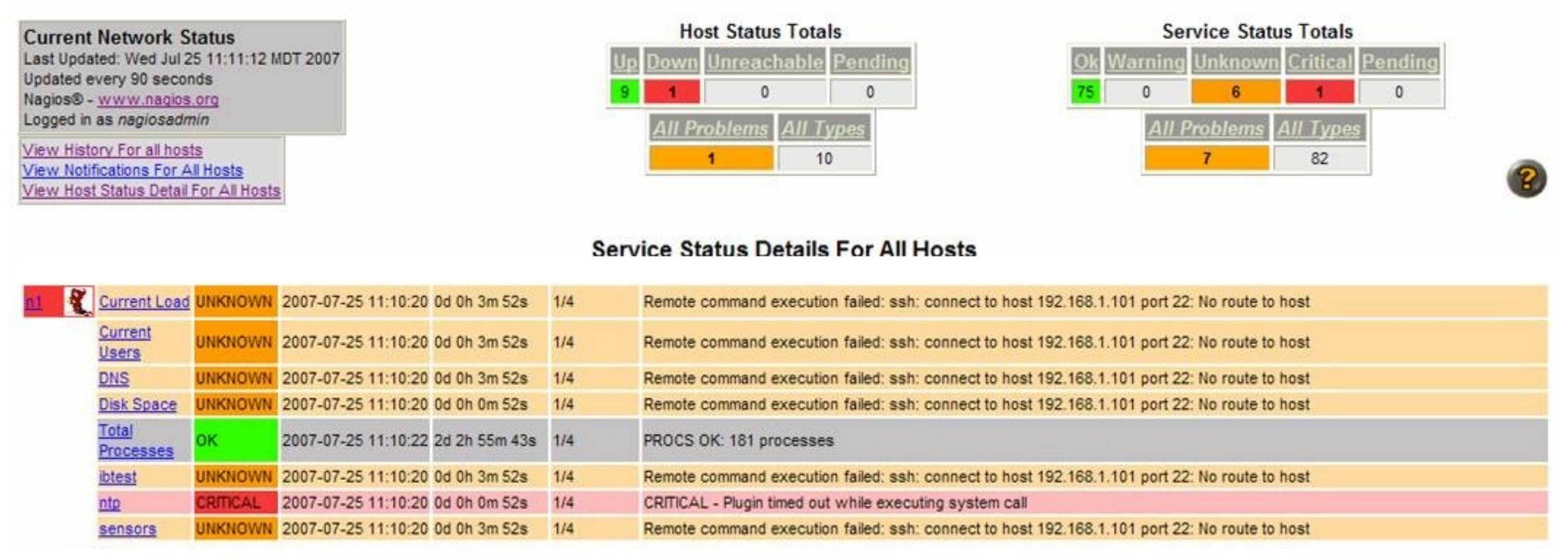
Nagios Detection:

In the event of critical errors and systems warnings, Nagios emails system administrators.
Detected several faulty fans in the system.

Failure Injection:

Manually stopped

Service checks
- Distributed monitoring and failover
- Custom plug-ins
- Multiple alert states
- Intelligently schedules service checks



Current Network Status
Last Updated: Wed Jul 25 11:11:12 MDT 2007
Updated every 90 seconds
Nagios® - www.nagios.org
Logged in as *nagiosadmin*
View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
9	1	0	0

All Problems	All Types
1	10

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
75	0	6	1	0

All Problems	All Types
7	82

Service Status Details For All Hosts

n1	Current Load	UNKNOWN	2007-07-25 11:10:20	0d 0h 3m 52s	1/4	Remote command execution failed: ssh: connect to host 192.168.1.101 port 22: No route to host
	Current Users	UNKNOWN	2007-07-25 11:10:20	0d 0h 3m 52s	1/4	Remote command execution failed: ssh: connect to host 192.168.1.101 port 22: No route to host
	DNS	UNKNOWN	2007-07-25 11:10:20	0d 0h 3m 52s	1/4	Remote command execution failed: ssh: connect to host 192.168.1.101 port 22: No route to host
	Disk Space	UNKNOWN	2007-07-25 11:10:20	0d 0h 0m 52s	1/4	Remote command execution failed: ssh: connect to host 192.168.1.101 port 22: No route to host
	Total Processes	OK	2007-07-25 11:10:22	2d 2h 55m 43s	1/4	PROCS OK: 181 processes
	ibtest	UNKNOWN	2007-07-25 11:10:20	0d 0h 3m 52s	1/4	Remote command execution failed: ssh: connect to host 192.168.1.101 port 22: No route to host
	ntp	CRITICAL	2007-07-25 11:10:20	0d 0h 0m 52s	1/4	CRITICAL - Plugin timed out while executing system call
	sensors	UNKNOWN	2007-07-25 11:10:20	0d 0h 3m 52s	1/4	Remote command execution failed: ssh: connect to host 192.168.1.101 port 22: No route to host

Conclusion:

Future Work:
More analysis of different monitoring methods and the amount of system noise and resource usage is is introduced by those methods is necessary. All monitoring requires system resources, but the optimum solution will provide accurate data while not negatively effecting the performance and execution of user code.